

# Python Installation Notes

Edition 10.2022

## Content

<b>1</b>	<b>Overview.....</b>	<b>2</b>
<b>2</b>	<b>Course Setup.....</b>	<b>3</b>
2.1	Miniconda.....	3
2.2	Visual Studio Code.....	4
2.3	Course Samples.....	6
2.4	Project Setup.....	7
2.5	Install Libraries .....	8
2.6	Install Jupiter Notebook.....	9
<b>3</b>	<b>Further Information.....</b>	<b>10</b>
3.1	Run Python Code .....	10
3.2	Run Python Code Snippet.....	10
3.3	Debug Python Code .....	11
3.4	Install and use Packages .....	12
3.5	Enable Unit Testing in VS Code.....	13
<b>4</b>	<b>Ubuntu Setup.....</b>	<b>14</b>
4.1	Miniconda.....	14
4.2	Visual Studio Code.....	15
4.3	GIT .....	16
4.4	Course Samples.....	16
4.5	Project Setup.....	16
<b>5</b>	<b>Appendix .....</b>	<b>16</b>
5.1	VS Code – Hints.....	16
5.2	VS Code – Terminal Setting.....	17



### 1 Overview

The following guide is just one option to get your Python setup ready for the course. If you have already a running Python environment with an appropriate IDE (Integrated Development Environment) you only need to download the Course Samples and check if the `hello.py` sample is running and install the libraries from Chapter 2.5 and 2.6.

If not, please follow the installation steps of **Chapter 2 Course Setup** and ensure, that the **hello.py** sample is **up** and **running**.

#### **Important**

**You need administrator rights for the installation.**

**We cannot support installation issues during the course.**



## 2 Course Setup

### 2.1 Miniconda

Miniconda is a free minimal installer for conda. It is a small, bootstrap version of Anaconda that includes only conda, Python, the packages they depend on, and a small number of other useful packages, including pip, zlib and a few others.

#### 1 - Download and install Miniconda for Python 3.8 or above:

<https://docs.conda.io/en/latest/miniconda.html>

Windows installers			
Windows			
Python version	Name	Size	SHA256 hash
Python 3.8	Miniconda3 Windows 64-bit	57.0 MiB	4fa22bba0497babb5b6608cb8843545372a99f5331c8120099ae1
	Miniconda3 Windows 32-bit	54.2 MiB	9c2ef76bae97246c85c206733ca30fd1feb8a4b3f90a2a511fea6

#### 2 - Settings during the installation:

- 1) Choose an installation location or use the default proposed by Miniconda
- 2) Enable Checkbox: Register Miniconda as my default Python
- 3) Finish Setup

# Python Installation Notes



## 2.2 Visual Studio Code

Install and configure Visual Studio Code.

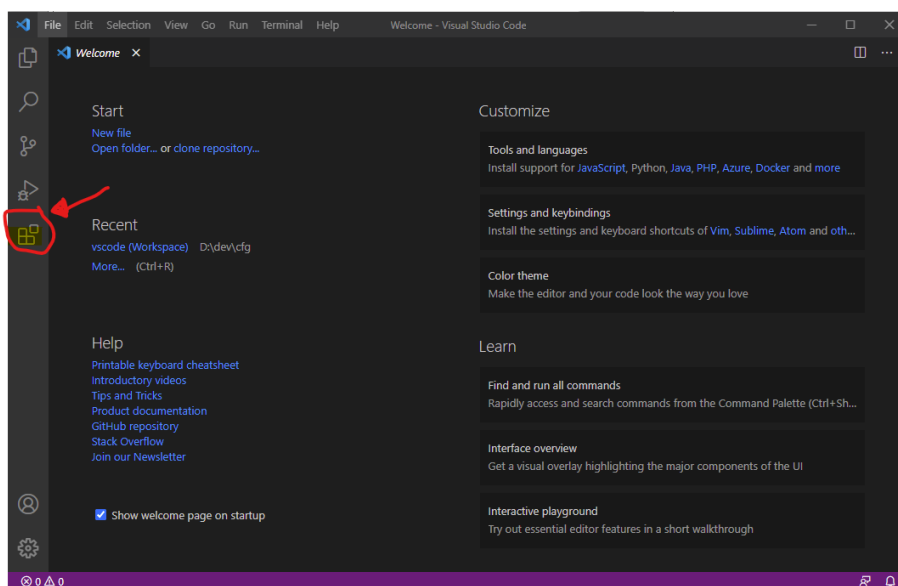
### 1 - Download and Install Visual Studio Code:

<https://code.visualstudio.com>

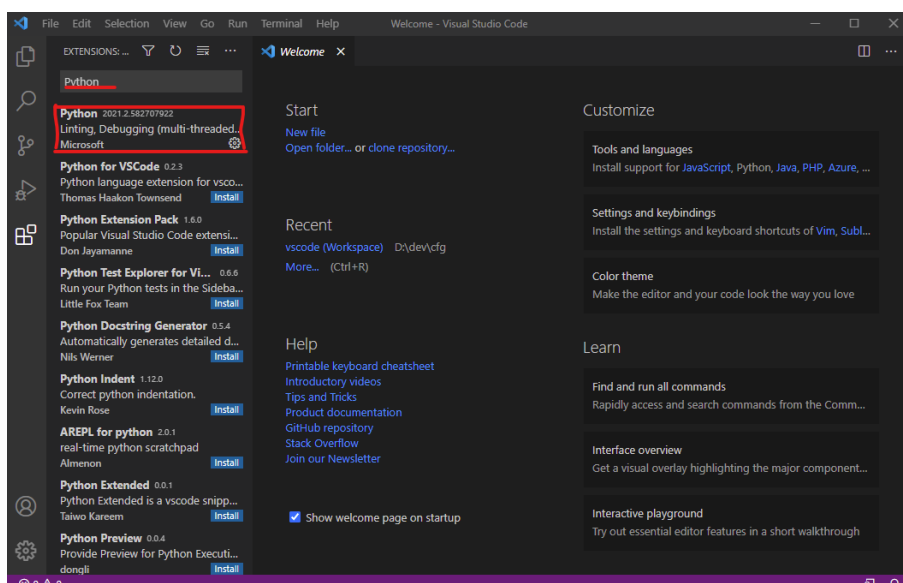
### 2 - Start VS Code and Install the Python Extension

### 3 – Install the Python Extension

On the left Side, Click the Plugin Button:



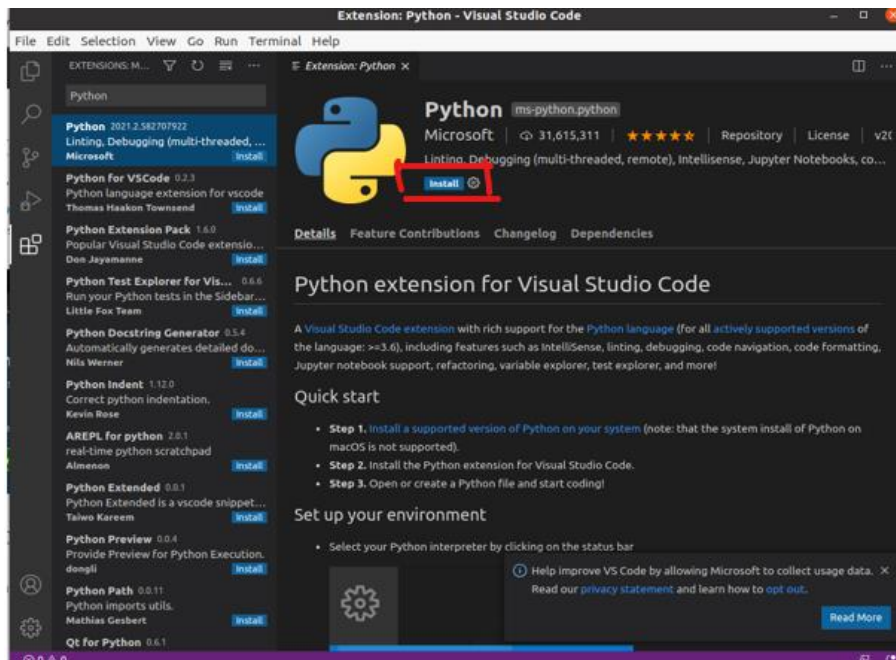
On the search field type: Python



## Python Installation Notes



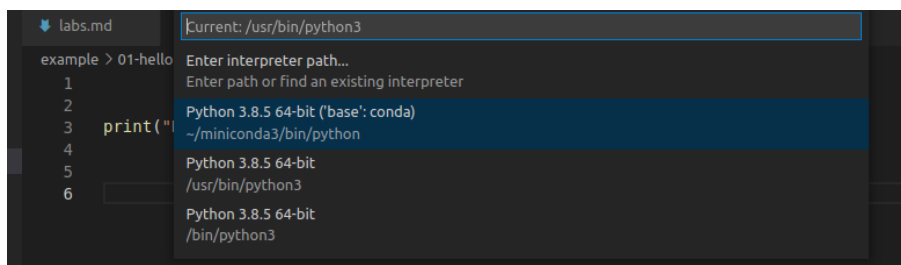
The Choose the **Python** Plugin and Click the **Install** Button



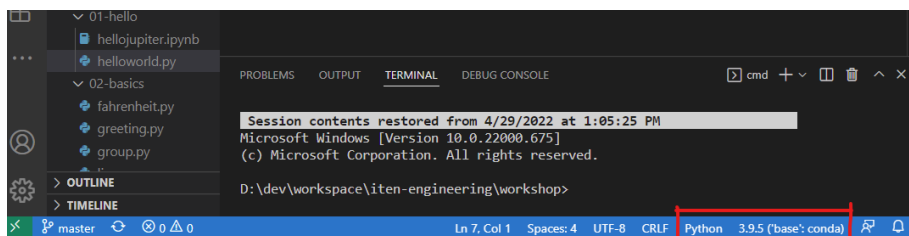
Note: This will also install the Jupiter Plugin for you.

### 4 - Select a Python interpreter:

1. Press: Ctrl+Shift+P
2. Enter: Python: Select Interpreter
3. Select: ~/miniconda... Installation



See Python interpreter Settings on the right side of the Bottom Bar:



### Further Information

For more details see: <https://code.visualstudio.com/docs/python/python-tutorial>



### 2.3 Course Samples

For the course we need the following GitHub project:

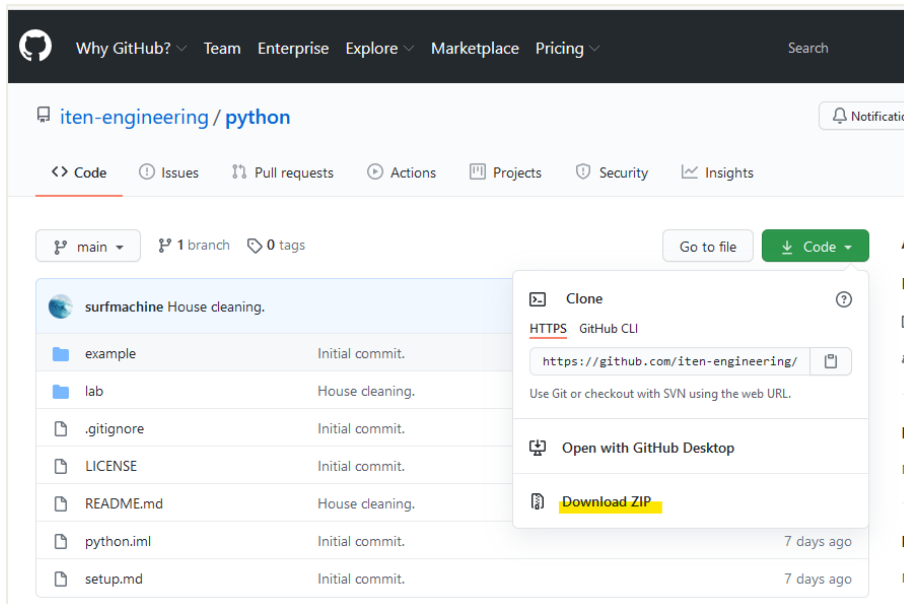
- <https://github.com/iten-engineering/python>

The project contains the examples of the slides, as well as all labs, we are doing during the course. **This is what we have to install now.**

You can either download a ZIP File or if you have GIT installed pull down the projects.

#### Option A – Download Course Samples as ZIP

<https://github.com/iten-engineering/python>



Download the ZIP File and extract the content to your working directory.

For example: `c:\course\workspace\python`

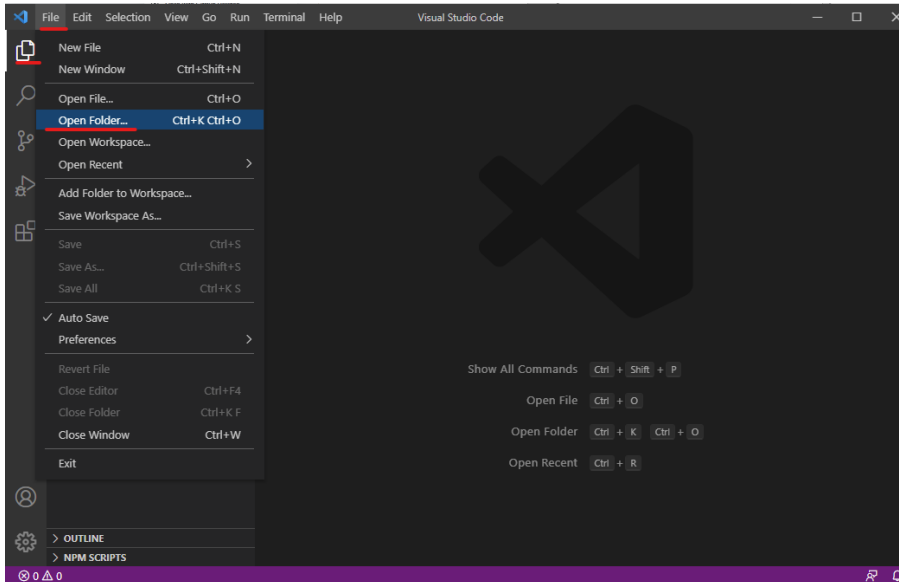
#### Option B – Install Projects with GIT

- Create a **workspace** directory for the course projects.
- Open the **Git Bash** and change to the workspace directory
- git clone <https://github.com/iten-engineering/python>



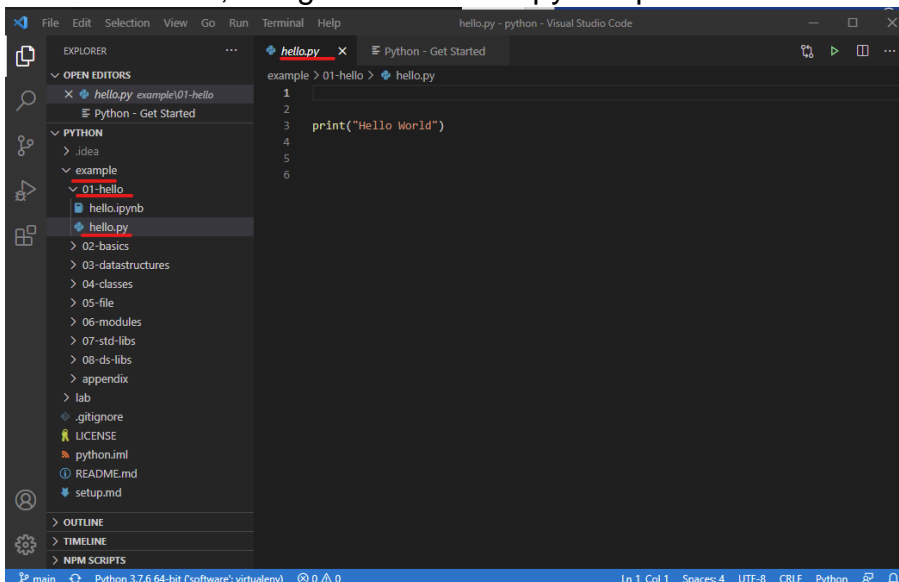
## 2.4 Project Setup

Start VS Code, Click the Explorer Button on the top left corner, then choose File / Open Folder:



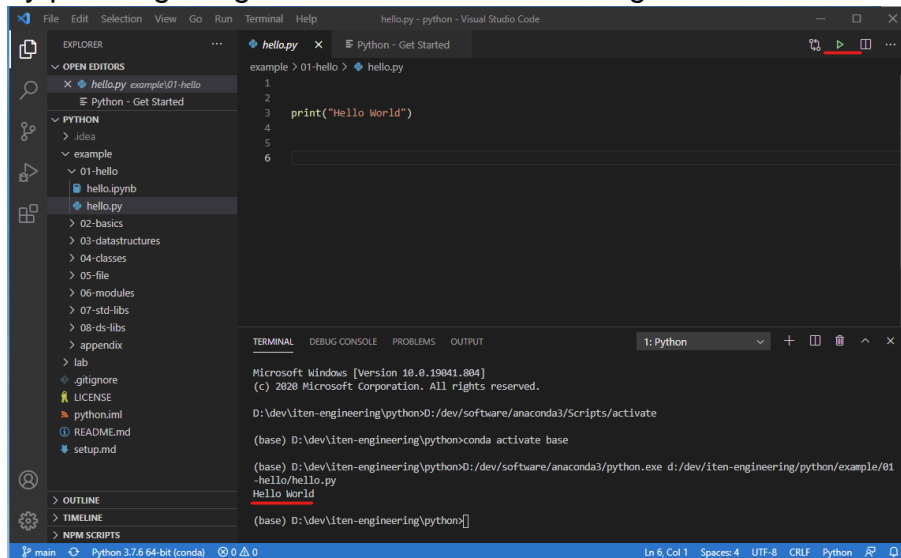
Navigate to your folder with the Course samples and select the folder.  
For example: `c:\course\workspace\python`

On the left side, navigate to the hello.py sample and click it:





Now you can execute the script,  
by pressing the green Arrow button on the right corner:



If your output looks similar to the Screenshot above, your fine.  
Congratulation to your first Hello World sample with Python!

If your output shows the following Error Message, please follow the steps described in **Appendix 5.2 “VS Code – Terminal Setting”**.

conda : The term 'conda' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again. At line:1 char:1

## 2.5 Install Libraries

For the advanced samples and showcases we need some more libraries. To have everything ready, please install the following libraries.

Open a Terminal in VS Code (or the Anaconda shell) and run the following commands:

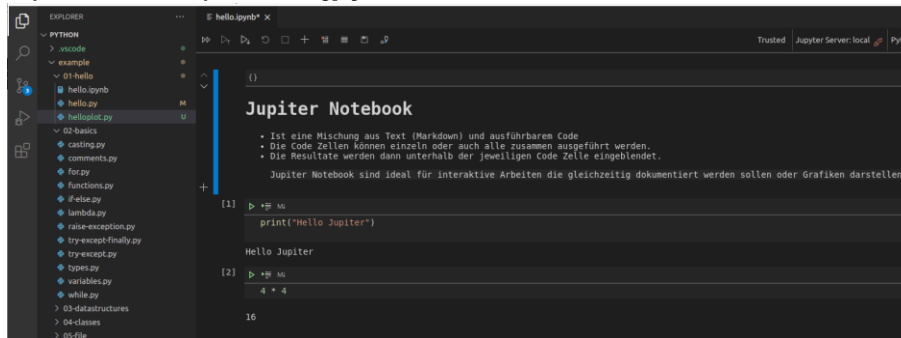
```
pip install matplotlib
pip install Flask
pip install -U scikit-learn
pip install streamlit
```



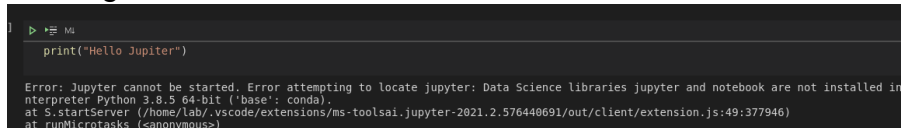


## 2.6 Install Jupiter Notebook

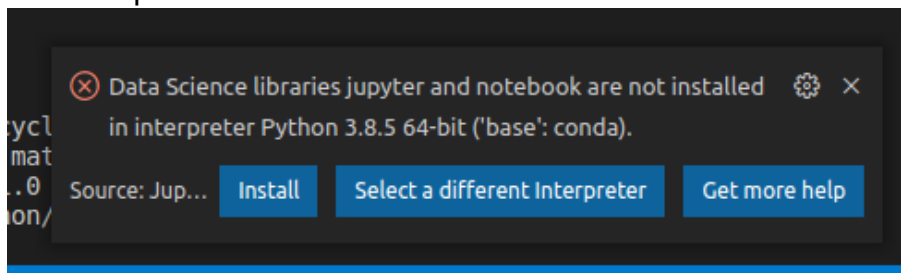
1. Open the Script **hello.jpynb**:



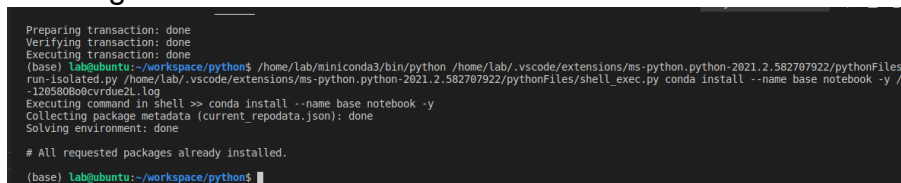
2. When running the first cell for the first time, you will see the following error message:



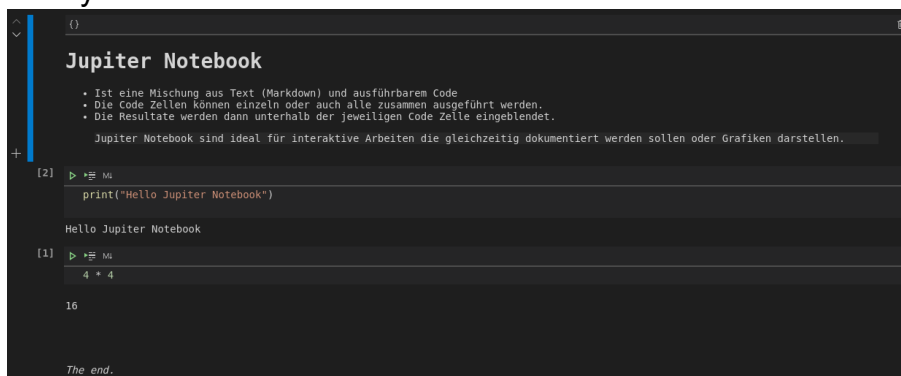
3. Install Jupyter: Click the “Install” Button:



Running installation:



4. Now you can run the cell's of the Notebook:





## 3 Further Information

This chapter contains further information we will use during the course. You do not need to follow them as part of the course preparation. If you still have some energy and time, feel free to go ahead.

### 3.1 Run Python Code

Create a new Script **hello.py** with the given content:

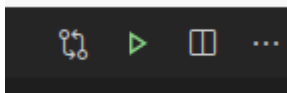
```

1
2
3
4 msg = "Hello World"
5 print(msg)
6

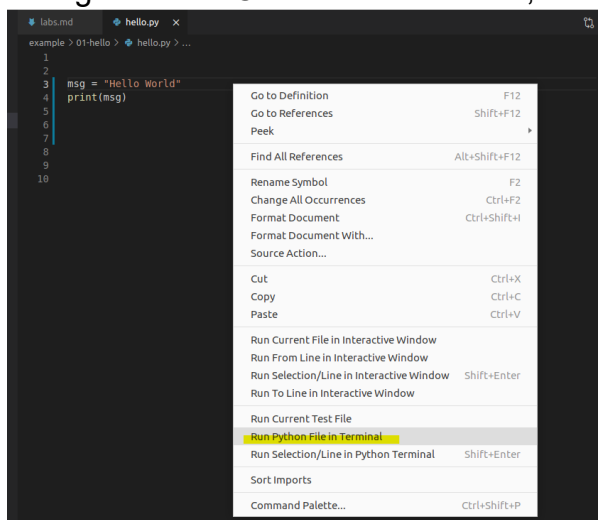
```

Run the python Script:

1. Press: Green Arrow on top/right



2. or Right Mouse Click in the File Area, then: "Run Python File in Terminal"



### 3.2 Run Python Code Snippet

Mark a portion of your Python Script.

Press: SHIFT + ENTER to run the marked code snippet



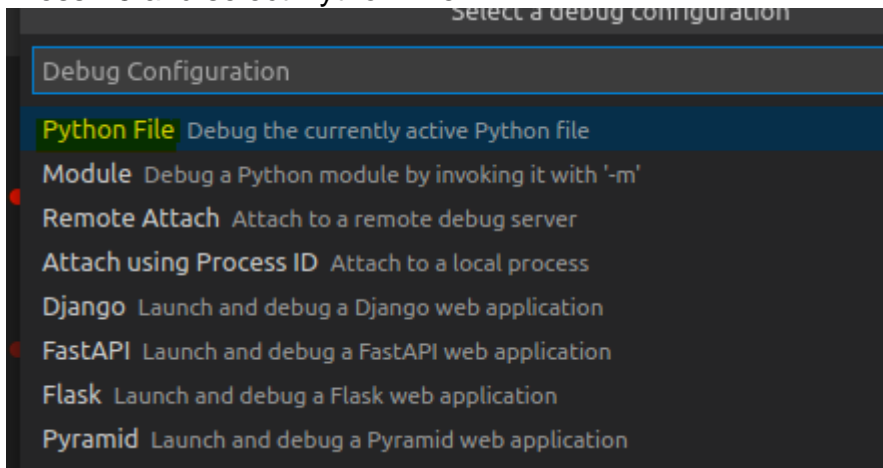
### 3.3 Debug Python Code

#### 1. Set Breakpoint

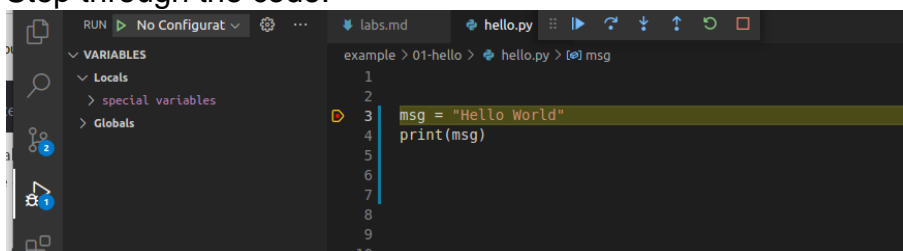
by pressing F9 on line 3 or click on the left side of the line number:

```
example > 01-hello > hello.py > ...  
1  
2  
3 msg = "Hello World"  
4 print(msg)  
5  
6  
7  
8  
9  
10
```

#### 2. Press F5 and select Python File:



#### 3. Step through the code:





### 3.4 Install and use Packages

1. Add new File **helloplot.py** with the following content:

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 20, 100)    # Create a list of evenly-spaced numbers over
the range

plt.plot(x, np.sin(x))        # Plot the sine of each x point
plt.show()                    # Display the plot
```

2. When running the code the first time you will see the following error message:

```
Traceback (most recent call last):
  File "/home/lab/workspace/python/example/01-hello/helloplot.py", line 1,
in <module>
    import matplotlib.pyplot as plt
ModuleNotFoundError: No module named 'matplotlib'
```

3. In the terminal with the error message, we can now install the missing libraries:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: Python
Hello World
(base) lab@ubuntu:~/workspace/python$ /home/lab/miniconda3/bin/python /home/lab/workspace/python/example/01-hello/hello.py
Hello World
(base) lab@ubuntu:~/workspace/python$ /home/lab/miniconda3/bin/python /home/lab/workspace/python/example/01-hello/helloplot.py
Traceback (most recent call last):
  File "/home/lab/workspace/python/example/01-hello/helloplot.py", line 1, in <module>
    import matplotlib.pyplot as plt
ModuleNotFoundError: No module named 'matplotlib'
(base) lab@ubuntu:~/workspace/python$
(base) lab@ubuntu:~/workspace/python$
(base) lab@ubuntu:~/workspace/python$ ls
example  lab  LICENSE  python.iml  README.md  setup.md
(base) lab@ubuntu:~/workspace/python$
```

Don't use with Anaconda distributions because they include matplotlib already.

**# macOS**

**python3 -m pip install matplotlib**

**# Windows (may require elevation)**

**python -m pip install matplotlib**

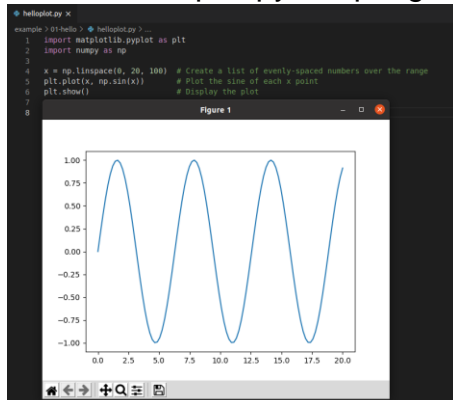
**# Linux (Debian)**

**sudo apt install python3-tk**

**python3 -m pip install matplotlib**



## 4. Run the helloplot.py Script again:



## 3.5 Enable Unit Testing in VS Code

Testing in Python is disabled by default. To enable testing, use the Python: Configure Tests command on the Command Palette. This command prompts you to select a test framework, the folder containing tests, and the pattern used to identify test files.

1. Shift+Ctrl+P
2. Enter: Python: Configure Tests
3. Choose **unittest**:

4. Choose folder: 04
5. Choose pattern: **\*test.py**



## 4 Ubuntu Setup

### 4.1 Miniconda

To install Miniconda on Ubuntu 20.04 from command line, it only takes 3 steps excluding creating and activating a conda environment.

1. **Download the latest shell script**

wget [https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86\\_64.sh](https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh)

2. **Make the miniconda installation script executable**

chmod +x Miniconda3-latest-Linux-x86\_64.sh

3. **Run miniconda installation script**

./Miniconda3-latest-Linux-x86\_64.sh

```
Miniconda3 will now be installed into this location:
```

```
/home/lab/miniconda3
```

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

```
[/home/lab/miniconda3] >>>  
PREFIX=/home/lab/miniconda3  
Unpacking payload ...
```

```
Preparing transaction: done  
Executing transaction: done  
installation finished.  
Do you wish the installer to initialize Miniconda3  
by running conda init? [yes|no]  
[no] >>> yes
```

```
==> For changes to take effect, close and re-open your current shell.<==
```

```
If you'd prefer that conda's base environment not be activated on startup,  
set the auto_activate_base parameter to false:
```

```
conda config --set auto_activate_base false
```

```
Thank you for installing Miniconda3!
```



## Miniconda commands

Command	Description
conda --version	Show version
conda info --envs	Show environments
conda create -n myenv python=3.6.2	Create environment with given python version
conda env remove -n myenv	Remove environment
conda activate myenv	Activate the myenv environment
conda list	Show packages of active environment
conda install [package name]	Install Package
conda update [package name]	Update Package

## 4.2 Visual Studio Code

Previous

Ready to go

Done

You're ready to go!

You can use "Software" to install apps like these:

Visual Studio Code

Skype

Spotify

Slack

PyCharm CE

Android Studio

IDEA Community

Krita

Plex Media Server

VLC

GNU Image Manipulation Program

GitKraken

Discord

Bombsquad

Bitwarden

Open "Software" now

Firefox Web Browser

Visual Studio Code

Code editing. Redefined.

★★★★★ (1162)

install

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Linux, macOS and Windows. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

By downloading and using Visual Studio Code, you agree to the license terms (<https://code.visualstudio.com/license/>) and privacy statement (<https://privacy.microsoft.com/en-us/privacystatement>). Visual Studio Code automatically sends telemetry data and crash dumps to help us improve the product. If you would prefer not to have this data sent please go see [https://code.visualstudio.com/docs/supporting/FAQ#\\_how-to-disable-crash-reporting](https://code.visualstudio.com/docs/supporting/FAQ#_how-to-disable-crash-reporting) to learn how to disable it.

VS Code <https://github.com/Microsoft/vscode>

For the configuration steps, follow the Course Setup chapter.



### 4.3 GIT

- **Install GIT**

sudo apt install git

```
(lab) lab@ubuntu:~/workspace$ sudo apt install git
[sudo] password for lab:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs
  git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 5,464 kB of archives.
After this operation, 38.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

- **Check Version**

git --version

- **Global Settings**

git config --global user.name "Pipi Langstrumpf"

git config --global user.email [pipi@villakunterbunt.org](mailto:pipi@villakunterbunt.org)

### 4.4 Course Samples

1. **Create Workspace Directory**

mkdir workspace

2. **Clone Python Project**

cd workspace

git clone <https://github.com/iten-engineering/python>

3. **Clone Workshop Project**

cd workspace

git clone <https://github.com/iten-engineering/workshop>

### 4.5 Project Setup

See chapter Course Setup.

## 5 Appendix

### 5.1 VS Code – Hints

Zoom:

- Menu: View / Appearance / Zoom-In      CTRL + NUM\_PAD\_ADD
- Menu: View / Appearance / Zoom-Out      CTRL + -

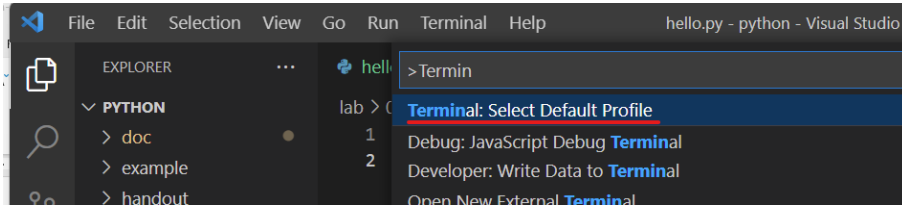




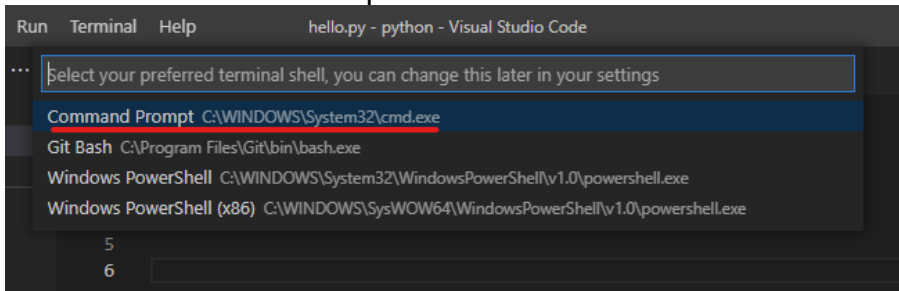
### 5.2 VS Code – Terminal Setting

1) Open the command palette with: **CTRL-SHIFT-P**

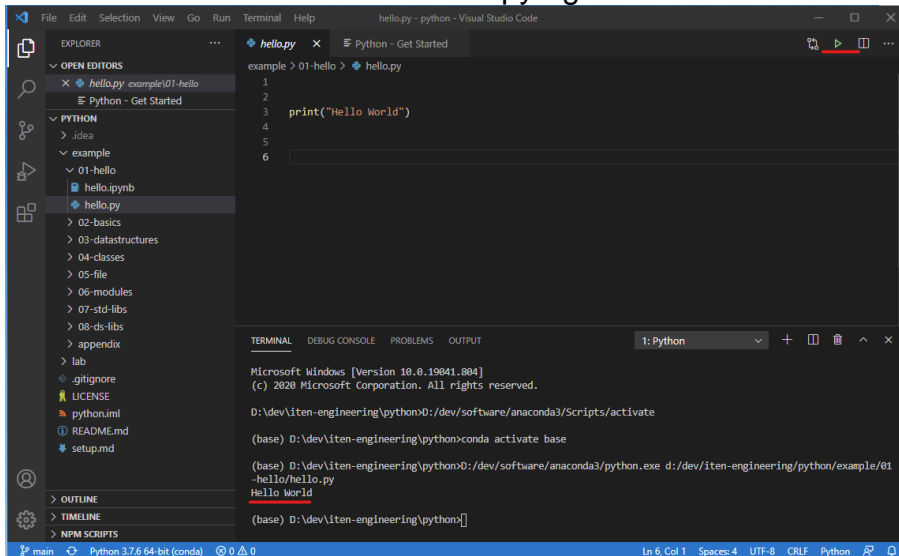
2) Search for **Terminal: Select Default Profile**



3) Choose: Command Prompt



4) Restart VS Code and run the hello.py again:



Details see: <https://stackoverflow.com/questions/54828713/working-with-anaconda-in-visual-studio-code>